Universidad de los Andes 28/05/2018

David Niño 201412734

María del Rosario León 201423755

Proyecto de Diseño y análisis de algoritmos problema A

* 1. Ecuación de recurrencia:
  2. Descripción de algoritmo: El algoritmo tiene 2 indices, el primero va limitando la parte derecha de la subcadena y el segundo va recorriendo el resto del arreglo para ir contando los ceros y asi determinar el tamaño de la subcadena más larga
     1. Método calculate():
        1. Contexto: nums: arreglo de números, c: número de ceros en la subcadena
        2. Precondición:
        3. Invariante:
        4. Post-condición:
     2. Método toNumArray():
        1. Contexto: split: arreglo de Strings
        2. Precondición:
        3. Invariante:
        4. Post-condición:
  3. Complejidad temporal: se recorre el arreglo 2 veces.
  4. Complejidad espacial: S(n) se necesita un arreglo adicional para calcular la sub-cadena.

1. Comentarios finales: Se considera que es una buena complejidad pues en teoría estamos evaluando todas las parejas posibles en O(1). Dada la situación se esperaría que la complejidad espacial fuera alta pero es posible optimizarla a un S(n) dado a que se hizo un análisis sobre el grafo de necesidades.